An Efficient Parallel Hierarchical Value Iteration Algorithm

N. HAFIDI*, A. LARACH*, and C. DAOUI*

*Laboratory of Information Processing and Decision Support

Sultan Moulay Slimane University, Faculty of science and Technique, Beni-Mellal, Morocco.

mr,na,hafidi@gmail.com

CBI' 2019

http://www.cbi-bm.com/

1. Introduction

This paper presents Parallel Value Iteration (PVI) algorithm for Discounted Markov decision processes and proposes an efficient Parallel Hierarchical Value Iteration (PHVI) based on the decomposition of the state space in to Strongly Connected Component (SCC) that can be classified into some levels. The proposed PHVI solver takes account the dimension of each SCC in each level for an efficient distribution of multithreading. Experimental results show the performance of the PHVI algorithm proposed.

2. Motivation and Background

The MDP theory [1, 2] has proven tremendously useful in a wide area of disciplines [3] and had received great attention from the Artificial Intelligence (AI) community since it offers easy frame work for modelling several complex real-life problems. Inventory Control, Maintenance Management, robot navigation, etc. Efficient solvers for MDPs had been developed, the most widely algorithms are Policy Iteration (PI), Value Iteration (VI) and its variants [1, 2]. Classical MDP algorithms are intractable to solve Markov Decision Process with large state and action spaces. It was a subject of several recent studies [4-5] to overcome the computational complexity, among them, reachability considerations , parallelism, aggregation and decomposition techniques.

An MDP solver can take advantages of the independences of calculates and the decomposition technique to parallelize them. Much of the recent work [4-5] has focused on accelerating the solver by using the parallelism and the decomposition technique into SCCs and there belonging level, but they don't take account the dimension of each SCC in each level which can lose the execution time.

4.1. Parallel Value Iteration Algorithm

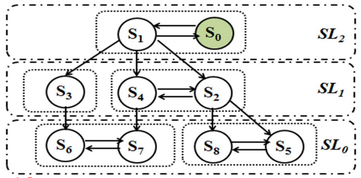
Let $\boldsymbol{N_P}$ be the number of processors and let consider the backup bellman equation:

$$\boldsymbol{V_t(i) = max_a\,[\,R(i,a) + \gamma \sum_{j\in E} P_a(i,a,j)\,V_{t-1}(i)\,]} \quad (1)$$

The values function $\boldsymbol{V_t(i)}$ of each state i are independent and can be calculated in parallel, the state space is therefore divided into $\boldsymbol{N_P}$ subset (S_1, S_2,…S_{Np}), a thread is responsible for calculating the value of each subset S_i .

4.2. Decomposition into SCC.

Let's consider the oriented graph G(S,U) associated to the original MDP, There exist a unique partition T = C_1, C_2,. . . , C_p of the state space S into SCCs, that can be classified into some levels.(Fig. 2).



4.3. Parallel Hierarchical Value Iteration Algorithm

After portioning the state space into **SCCs** and constructing the levels, an efficient number of processors is calculated for each **SCCs** in each level, than the **PVI** algorithm is applied.

3. Methodology

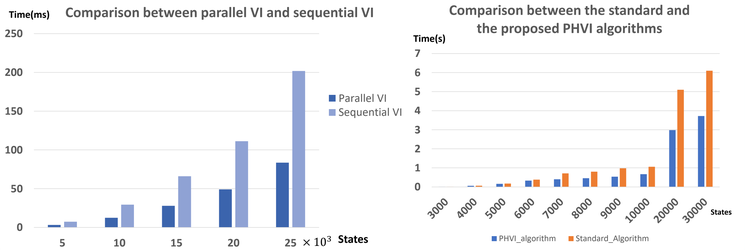
PVI (MDP: S, A, P, R, N_P : number of processors)

1. **Divide** S into $\boldsymbol{N_P}$ partitions S_1, S_2,…S_{Np}
2. **While** (Not convergence) **do**
   - **For each** partition $\boldsymbol{S_{p_k}}$, k=1,…,N_P **Do in parallel**
     $$\boldsymbol{V_n(i) = max_a\,[\,R(i,a) + \alpha \sum_{j\in N_{p_k}} P_{iaj}\,V_{n-1}(i)\,]} \quad (1)$$
3. **For each** partition $\boldsymbol{N_{p_k}}$, k=1,…, N_P **Do in parallel**
   $$\boldsymbol{\pi^*(i) = max_a\,[\,R(i,a) + \alpha \sum_{j\in N_{p_k}} P_{iaj}\,V_n(i)\,]} \quad (2)$$

PHVI (MDP: S, A, P, R, N_P : number of processors)

1. **Construct** Levels using Tarjan's algorithm
2. **For each** *Level* L_k $k = 0$ *to* L ***Do***
   - $N_L \leftarrow 0$ //Number of states in each level
   - **For each** SCC_i $i = 0$ *to* N_{SCC} **Do**
     $$N_L \leftarrow N_L + N_{SCC_i} \quad //Number\ of\ stats\ in\ SCC_i$$
     $$N_a = \frac{N_L}{N_p} \quad //\text{Average number of state in each SCC}$$
   - **For each** SCC_i $i = 0$ *to* N_{SCC} **Do In Parallel**
     $$N_{P_i} = \frac{N_L}{N_a} \quad //Number\ of\ processors\ affected\ to\ the\ SCC_i$$
     $$PVI(MDP_k,\ N_{P_i})$$

4. Results



5. Conclusions and Future Work

In this paper, we have proposed an efficient Parallel Hierarchical Value Iteration Algorithm tacking account the size of each **SCC** in each level, the experimental results show that the proposed solver outperforms the standard **PHVI** algorithm. In future works, we try to apply this approach to the Policy Iteration algorithm and to the distributed case.

References

[1] R. A. Howard. Dynamic Programming and Markov Processes. MITPress, Cambridge, Massachusetts, 1960.

[2] M. Puterman. Markov Decision Process: Discrete stochastic Dynamic Programming. John Willey & Sons, Inc, New York, 1994.

[3] M.L.Littman, T.L.Dean, and L.P.Kaelbling, "On the complexity of solving Markov decision problems." In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1995. p. 394-402.

[4] A. Larach, S. Chafik and C. Daoui, Accelerated decomposition techniques for large Discounted Markov decision processes. J.Ind.Eng.Int,13:417- 426, 2017.

[5] Chafik S, Daoui C (2015) A modified value iteration algorithm for discounted Markov decision process. J Electron Commerce Organ 13(3):47–57.